ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

Distributed Primal Decomposition for Large-Scale MILPs

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:
Camisa A., Notarnicola I., Notarstefano G. (2022). Distributed Primal Decomposition for Large-Scale MILPs. IEEE TRANSACTIONS ON AUTOMATIC CONTROL, 67(1), 413-420 [10.1109/TAC.2021.3057061].

Availability:
This version is available at: https://hdl.handle.net/11585/818692 since: 2021-06-16

Published:
DOI: http://doi.org/10.1109/TAC.2021.3057061

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/).
When citing, please refer to the published version.

(Article begins on next page)

16 November 2026

This is the final peer-reviewed accepted manuscript of:

A. Camisa, I. Notarnicola and G. Notarstefano, “Distributed Primal Decomposition for Large-Scale MILPs”, in IEEE Transactions on Automatic Control

The final published version is available online at:

http://dx.doi.org/10.1109/TAC.2021.3057061

Rights / License:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/)

When citing, please refer to the published version.

Distributed Primal Decomposition for Large-Scale MILPs

Andrea Camisa, *IEEE Student Member*, Ivano Notarnicola, *IEEE Member*, Giuseppe Notarstefano, *IEEE Member*

***Abstract*—This paper deals with a distributed Mixed-Integer Linear Programming (MILP) set-up arising in several control applications. Agents of a network aim to minimize the sum of local linear cost functions subject to both individual constraints and a linear coupling constraint involving all the decision variables. A key, challenging feature of the considered set-up is that some components of the decision variables must assume integer values. The addressed MILPs are NP-hard, nonconvex and large-scale. Moreover, several additional challenges arise in a distributed framework due to the coupling constraint, so that feasible solutions with guaranteed suboptimality bounds are of interest. We propose a fully distributed algorithm based on a primal decomposition approach and an appropriate tightening of the coupling constraint. The algorithm is guaranteed to provide feasible solutions in finite time. Moreover, asymptotic and finite-time suboptimality bounds are established for the computed solution. Montecarlo simulations highlight the extremely low suboptimality bounds achieved by the algorithm.**

***Index Terms*—Distributed Optimization, Mixed-Integer Linear Programming, Constraint-Coupled Optimization**

I. Introduction

In this paper, we investigate large-scale Mixed-Integer Linear Programs (MILPs) that are to be solved by a network of agents without a central coordinator. The goal is to minimize the sum of objective functions while satisfying individual constraints and a common, non-sparse coupling constraint. We term these MILPs *constraint coupled*. Due to the mixed-integer decision variable, the large-scale size and the coupling constraint, these problems turn out to be extremely challenging in a distributed context. This typically arise in several relevant control applications, such as microgrid control, economic dispatch in power systems, task assignment in cooperative robotics. An interesting scenario arises in distributed Model Predictive Control (MPC), where a large set of nonlinear control systems must cooperatively solve a common control task and their states, outputs and/or inputs are coupled through a coupling constraint. Here, the constraint-coupled structure results directly from the problem formulation, and the integrality constraints stem from the MILP approximation of the original optimal control problem [2]. In cooperative MPC schemes, such a complex optimization problem should be ideally solved at each control iteration (see e.g. [3]–[7]). Being these problems NP-hard, it is not computationally affordable to achieve exact optimality, however feasible (suboptimal) solutions are often sufficient to guarantee stability. It is thus of great interest to compute "good-quality" feasible solutions of large-scale MILPs.

A preliminary short version of this paper has appeared as [1]. The current article provides a more detailed discussion, an improved algorithm with tighter restriction, all the theoretical proofs and an extensive numerical study.

A. Camisa, I. Notarnicola and G. Notarstefano are with the Department of Electrical, Electronic and Information Engineering, University of Bologna, Bologna, Italy. {a.camisa, ivano.notarnicola, giuseppe.notarstefano}@unibo.it. This result is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 638992 - OPT4SMART).

Since our paper deals with constraint-coupled optimization, we organize the relevant literature in two parts. First, we review existing methods for convex constraint-coupled problems. In the tutorial paper [8], parallel decomposition techniques are reviewed. A distributed gradient descent method is proposed in [9] to solve smooth resource allocation problems. In [10] a regularized saddle-point algorithm for convex optimization problems over networks is analyzed. In [11], [12] distributed algorithms based on Laplacian-gradient dynamics are used to solve economic dispatch over digraphs. In [13], [14] distributed dual decomposition-based algorithms for constraint-coupled problems are analyzed, while [15] proposes a distributed algorithm based on successive duality steps. Approaches for constraint coupled problems based on augmented Lagrangian methods with consensus schemes are investigated in [16], [17]. Finally, a discussion on approaches for convex constraint-coupled problems can be found in the recent survey paper [18].

Second, we review parallel and distributed algorithms for MILPs. In [19] a Lagrange relaxation approach is applied to demand response control in smart grids. In [20] a heuristic for embedded mixed-integer programming is studied to obtain approximate solutions. A first attempt to obtain a distributed approximate solution for MILPs is [21]. Recently, a distributed algorithm has been investigated in [22] to solve a different class of MILPs with shared decision variable. A pioneering work on fast, master-client parallel algorithms to find approximate solutions of our problem set-up is [23]. In [24], an enhanced version has been proposed to improve the quality of the solution. A distributed implementation of [24] is proposed in [25].

The contributions of this paper are as follows. We first focus on constraint-coupled convex problems and provide a distributed algorithm based on a combined relaxation and primal decomposition approach. Thanks to this first analysis, as second and main contribution of the paper, we then propose a distributed optimization algorithm for the fast computation of feasible solutions to large-scale, constraint-coupled MILPs. Our algorithm builds on the distributed primal decomposition applied to a convex approximation of the target MILP with restricted coupling constraint. For the mixed-integer solution estimate computed by the proposed distributed method, we are able to: (i) establish both asymptotic and finite-time feasibility, and (ii) provide both asymptotic and finite-time suboptimality bounds. Thanks to the primal decomposition reformulation, the proposed restriction of the coupling constraint turns out to be tighter than the state of the art. Through an extensive numerical study on randomly generated MILPs, we show that our approach is able to achieve interestingly low suboptimality gaps.

The paper unfolds as follows. In Section II, we introduce the MILP set-up together with useful preliminaries. In Section III, we propose our distributed algorithm which is analyzed in Section IV. In Section V, a numerical study is presented. All the proofs of the theoretical results are deferred to the appendix.

II. Optimization Set-up and Preliminaries

In this section, we introduce the problem set-up together with some preliminaries that act as building blocks for the development of our methodology.

A. Constraint-Coupled Mixed-Integer Linear Program

Let us consider a network of N agents aiming to solve the optimization problem

$$\begin{aligned} \min_{x_1,\dots,x_N} \ & \sum_{i=1}^{N} c_i^\top x_i \\ \text{subj. to } & \sum_{i=1}^{N} A_i x_i \le b \\ & x_i \in X_i^{\text{MILP}}, \qquad\qquad i \in \{1,\dots,N\}, \end{aligned} \tag{1}$$

where, for all $i \in \{1,\dots,N\}$, the decision variable x_i has p_i+q_i components and the mixed-integer constraint set is of the form $X_i^{\text{MILP}} = P_i \cap (\mathbb{Z}^{p_i} \times \mathbb{R}^{q_i})$, for some nonempty compact polyhedron $P_i \subset \mathbb{R}^{p_i+q_i}$. The decision variables are intertwined by S linear *coupling* constraints, described by the matrices $A_i \in \mathbb{R}^{S\times(p_i+q_i)}$ and the vector $b \in \mathbb{R}^S$. We assume that problem (1) is feasible and denote by $(x_1^\star,\dots,x_N^\star)$ an optimal solution with cost $J^{\text{MILP}} = \sum_{i=1}^N c_i^\top x_i^\star$. In many control applications, the number of decision variables is typically much larger than the number of coupling constraints. Therefore, in this paper we let $N \gg S$, leading to *large-scale* instances of problem (1).

We assume each agent i has a *partial knowledge* of problem (1), i.e., it knows only its local data X_i^{MILP}, c_i, A_i and b. The goal for each agent is to compute its portion $x_i^\star$ of an optimal solution of (1), by means of neighboring communication and local computation. Agents communicate according to a connected and undirected graph $\mathcal{G} = (\{1,\dots,N\},\mathcal{E})$, where $\mathcal{E} \subseteq \{1,\dots,N\}^2$ is the set of edges. The set of neighbors of i in $\mathcal{G}$ is $\mathcal{N}_i = \{j \in \{1,\dots,N\} \mid (i,j) \in \mathcal{E}\}$.

To solve MILP (1), one can employ enumeration schemes, such as branch-and-bound or cutting-plane techniques. However, this would not exploit its separable structure and would result into a computationally intensive algorithm. Therefore, in the next subsection we introduce an approximate version of the problem that preserves its structure while allowing for the application of efficient decomposition techniques.

B. Linear Programming Approximation of the Target MILP

Following [23]–[26], let us consider a modified version of problem (1) where the right-hand side of the coupling constraint is restricted by a vector $\sigma \ge 0$ and each mixed-integer set X_i^{MILP} is replaced by its convex hull, denoted by $\text{conv}(X_i^{\text{MILP}})$. The following convex problem is obtained

$$\begin{aligned} \min_{z_1,\dots,z_N} \ & \sum_{i=1}^{N} c_i^\top z_i \\ \text{subj. to } & \sum_{i=1}^{N} A_i z_i \le b - \sigma \\ & z_i \in \text{conv}(X_i^{\text{MILP}}), \qquad i \in \{1,\dots,N\}, \end{aligned} \tag{2}$$

where $z_i \in \mathbb{R}^{p_i+q_i}$ for all $i \in \{1,\dots,N\}$. We introduced z_i to clearly distinguish continuous variables from their mixed-integer counterparts in problem (1). When $\sigma = 0$, problem (2) is a relaxation of problem (1) and preserves its feasibility. When $\sigma > 0$, as done in related approaches [23]–[25], feasibility of problem (2) must be also assumed (see Assumption IV.1).

The main point in solving the (convex) problem (2) in place of the (nonconvex) original MILP (1) is to reconstruct a feasible solution of (1) starting from a solution of (2). The restriction σ is designed to guarantee that the solution is feasible for the coupling constraint and depends on the specific mixed-integer reconstruction procedure. Due to the feasibility assumption, the larger is σ, the narrower is the class of problems to which the approach can be applied. Notably, our method is no worse than [23] in terms of needed σ. In fact, numerical experiments highlight an extremely lower restriction magnitude, which, as a byproduct, results also in much less suboptimality of the computed solution.

Next, we introduce a key result based on the Shapley-Folkman lemma.

Proposition II.1. *Let problem* (2) *be feasible and let* $(\bar z_1,\dots,\bar z_N)$ *be any vertex of its feasible set. Then, there exists an index set* $I_{\mathbb{Z}} \subseteq \{1,\dots,N\}$*, with cardinality* $|I_{\mathbb{Z}}| \ge N - S$*, such that* $\bar z_i \in X_i^{\text{MILP}}$ *for all* $i \in I_{\mathbb{Z}}$*.* □

A proof is given in the early reference [26] (see also [23] for a more recent proof). Since Proposition II.1 gives a bound on the number of portions that are not mixed integer, the leading idea is to adapt only these portions while keeping the others intact. The approach will heavily rely on a primal decomposition framework whose foundations are reviewed in the next subsection.

C. Primal Decomposition Review

Primal decomposition is a powerful tool to recast constraint-coupled convex programs such as (2) into a master-subproblem architecture [27]. The right-hand side vector $b-\sigma$ of the coupling constraint is interpreted as a given (limited) resource to be shared among agents. Thus, local *allocation vectors* $y_i \in \mathbb{R}^S$ for all i are introduced such that $\sum_{i=1}^N y_i = b - \sigma$. To determine the allocations, a *master problem* is introduced

$$\begin{aligned} \min_{y_1,\dots,y_N} \ & \sum_{i=1}^{N} p_i(y_i) \\ \text{subj. to } & \sum_{i=1}^{N} y_i = b - \sigma \\ & y_i \in Y_i, \qquad\qquad i \in \{1,\dots,N\}, \end{aligned} \tag{3}$$

where, for each $i \in \{1,\dots,N\}$, $p_i : \mathbb{R}^S \to \mathbb{R}$ is defined as the optimal cost of the i-th (linear programming) *subproblem*

$$\begin{aligned} p_i(y_i) = \min_{z_i} \ & c_i^\top z_i \\ \text{subj. to } & A_i z_i \le y_i \\ & z_i \in \text{conv}(X_i^{\text{MILP}}). \end{aligned} \tag{4}$$

In problem (3), the constraint $Y_i \subseteq \mathbb{R}^S$ is the set of y_i for which problem (4) is feasible, i.e., such that there exists $z_i \in \text{conv}(X_i^{\text{MILP}})$ satisfying the local *allocation constraint* $A_i z_i \le y_i$. The following lemma ([27, Lemma 1]) establishes the equivalence between LP (2) and problems (3)–(4).

Lemma II.2. *Let* (2) *be feasible. Then, (i) the optimal costs of problems* (2) *and* (3) *are equal; (ii) if* $(y_1^\star, \dots, y_N^\star)$ *is optimal for* (3) *and, for all* i, $z_i^\star$ *is optimal for* (4) *(with* $y_i = y_i^\star$*), then* $(z_1^\star, \dots, z_N^\star)$ *is an optimal solution of* (2). □

Note that, given an optimal allocation $(y_1^\star, \dots, y_N^\star)$, each node can retrieve its portion of an optimal solution of problem (2) by relying only on its local allocation $y_i^\star$.

III. Distributed Primal Decomposition for MILPs

In this section we propose a novel distributed algorithm to compute a feasible solution of MILP (1). A cornerstone of the proposed strategy is the distributed primal decomposition method to solve the convex problem (2). We first present this scheme. Then, we formally describe our algorithm for MILPs together with its underlying rationale.

A. Distributed Primal Decomposition for Convex Problems

As already discussed, agents can compute the solution to (2) by independently solving problem (4), provided that an optimal allocation $(y_1^\star, \dots, y_N^\star)$ is available. To compute such optimal allocation, one can think of applying a projected subgradient method to problem (3). By denoting $p(y) = \sum_{i=1}^N p_i(y_i)$ the cost function of problem (3) and by letting $t \geq 0$ be an iteration index, the update reads $y^{t+1} = [y^t - \alpha^t \nabla p(y^t)]_{\mathcal{Y}}$, where α^t is the step-size and $[\cdot]_{\mathcal{Y}}$ denotes the Euclidean projection onto the feasible set of problem (3). The constraints $y_i \in Y_i$ in $\mathcal{Y}$ can be handled by resorting to a relaxation approach similar to the one used in [15]. As for the constraint $\sum_{i=1}^N y_i = b - \sigma$, the projection admits a closed-form solution that reads

$$y_i^{t+1} = y_i^t - \frac{\alpha^t}{N} \sum_{j=1}^N \big(\nabla p_i(y_i^t) - \nabla p_j(y_j^t)\big), \qquad \forall\, i \tag{5}$$

where α^t is the step-size. However, the update (5) requires knowledge of the entire vector $(\nabla p_1(y_1^t), \dots, \nabla p_N(y_N^t))$ and therefore cannot be performed in a distributed way. We next present an effective approach that bridges this gap.

Each agent i maintains a local allocation estimate y_i^t. At each iteration $t \geq 0$, agents compute μ_i^t as a Lagrange multiplier of

$$\begin{aligned} \min_{z_i, v_i} \quad & c_i^\top z_i + M v_i \\ \text{subj. to} \quad & \mu_i : \ A_i z_i \leq y_i^t + v_i \mathbb{1} \\ & z_i \in \text{conv}(X_i^{\text{MILP}}), \ \ v_i \geq 0, \end{aligned} \tag{6}$$

where $M > 0$ and $\mathbb{1}$ is the vector of ones. Then, each agent i receives μ_j^t from its neighbors $j \in \mathcal{N}_i$ and updates y_i^t with

$$y_i^{t+1} = y_i^t + \alpha^t \sum_{j \in \mathcal{N}_i} \big(\mu_i^t - \mu_j^t\big), \tag{7}$$

where α^t is the step-size. Note that problem (6) is a modified version of (4) that is feasible for all y_i and that $\nabla p_i(y_i^t) = \mu_i^t$ [28, Section 5.4.4]. It is introduced to take into account the constraints $y_i \in Y_i$ in (3). Intuitively, update (7) is obtained by making the centralized update (5) match the graph sparsity. As for the step-size in (7), the following standard assumption is made.

Assumption III.1. *The step-size sequence* $\{\alpha^t\}_{t\geq 0}$*, with each* $\alpha^t \geq 0$*, satisfies* $\sum_{t=0}^\infty \alpha^t = \infty$, $\sum_{t=0}^\infty \left(\alpha^t\right)^2 < \infty$. □

Next, we formalize the convergence result properties of the distributed primal decomposition algorithm for convex problems described by (6)–(7).

Proposition III.2. *Let Assumption III.1 hold. Moreover, let problem* (2) *be feasible and let the local allocation vectors* y_i^0 *be initialized such that* $\sum_{i=1}^N y_i^0 = b - \sigma$*. Then, for a sufficiently large* $M > 0$*, the distributed algorithm* (6)–(7) *generates an allocation vector sequence* $\{y_1^t, \dots, y_N^t\}_{t\geq 0}$ *and a primal sequence* $\{z_1^t, \dots, z_N^t\}_{t\geq 0}$ *(solutions of problem* (6) *for all* $t \geq 0$*) such that*

(i) $\sum_{i=1}^N y_i^t = b - \sigma$*, for all* $t \geq 0$*;*
(ii) $\lim_{t\to\infty} \|y_i^t - y_i^\star\| = 0$ *for all* $i \in \{1, \dots, N\}$*, where* $(y_1^\star, \dots, y_N^\star)$ *is an optimal solution of* (3)*;*
(iii) every limit point of $\{z_1^t, \dots, z_N^t\}_{t\geq 0}$ *is an optimal (feasible) solution of problem* (2). □

The proof relies on a primal decomposition reinterpretation of the algorithm in [15] and is omitted for space constraints. Operatively, the parameter $M > 0$ can be chosen by using an iterative update scheme as proposed in [26].

B. Distributed Algorithm Description

We are now ready to formally introduce our Distributed Primal Decomposition for MILPs. Each agent i maintains a mixed-integer solution estimate $x_i^t \in X_i^{\text{MILP}}$ and a local allocation estimate $y_i^t \in \mathbb{R}^S$. At each iteration $t \geq 0$, the local allocation estimate y_i^t is updated according to (6)–(7). After $T_f > 0$ iterations, agents compute a tentative mixed-integer solution $x_i^{T_f}$, based on the last computed allocation $y_i^{T_f}$ (cf. (8)). Here the notation lex-min denotes that ρ_i, ξ_i and x_i are minimized in a lexicographic order [18]. In Section III-C, we discuss in more detail the meaning of problem (8) and an operative way to solve it. Algorithm 1 summarizes the steps from the perspective of agent i.

Algorithm 1 Distributed Primal Decomposition for MILPs

Initialization: Set $T_f > 0$, y_i^0 such that $\sum_{i=1}^N y_i^0 = b - \sigma$

Evolution: For $t = 0, 1, \dots, T_f$ **do**

Compute μ_i^t as a Lagrange multiplier of (6)

Receive μ_j^t from $j \in \mathcal{N}_i$ and update y_i^{t+1} with (7)

Return $x_i^{T_f}$ as optimal solution of

$$\begin{aligned} \underset{\rho_i, \xi_i, x_i}{\text{lex-min}} \ & \rho_i \\ \text{subj. to} \ & c_i^\top x_i \leq \xi_i \\ & A_i x_i \leq y_i^{T_f} + \rho_i \mathbb{1} \\ & x_i \in X_i^{\text{MILP}}, \ \ \rho_i \geq 0. \end{aligned} \tag{8}$$

A sensible choice for y_i^0 is $y_i^0 = (b - \sigma)/N$. By Proposition III.2, the local allocation vectors $\{y_1^t, \dots, y_N^t\}_{t\geq 0}$ converge asymptotically to an optimal solution $(y_1^\star, \dots, y_N^\star)$ of problem (3). Moreover, owing to Proposition II.1, the asymptotic solution $z_i^\star$ of problem (6) is already mixed integer for at least $N - S$ agents. As for the remaining (at most) S agents, a recovery procedure is needed to guarantee that they also have have a mixed-integer solution. This is done via step (8). In

order to allow for a premature halt of the algorithm, we let *all the agents* perform (8). In Section IV, an asymptotic and finite-time analysis of Algorithm 1 is provided.

From an implementation point of view, an explicit description of $\text{conv}(X_i^{\text{MILP}})$ in terms of inequalities might not be available. Nevertheless, agents can still obtain an estimate of μ_i^t by *locally* running a dual subgradient method on (6), which involves the solution of small (local) MILPs without needing $\text{conv}(X_i^{\text{MILP}})$. The main limitation of the algorithm is only due to the local computation capability of each node.

C. Discussion on Mixed-Integer Solution Recovery

In this subsection, we describe in more detail the approach behind problem (8) and how it allows agents to recover a "good" solution of MILP (1). We first describe the procedure at steady state and then show how we cope with the finite number of iterations T_f.

Let $(z_1^\star, \dots, z_N^\star)$ be an optimal solution of the approximate problem (2) and let $(y_1^\star, \dots, y_N^\star)$ be a corresponding allocation of the master problem (3), computed asymptotically by Algorithm 1. A straight approach to recover a mixed-integer solution would be to solve for all i the optimization problem

$$\begin{aligned} \min_{x_i} \ & c_i^\top x_i \\ \text{subj. to } & A_i x_i \le y_i^\star \\ & x_i \in X_i^{\text{MILP}}. \end{aligned} \tag{9}$$

Problem (9) is a (small) local MILP that is reminiscent of the subproblem (4). Depending on the allocation constraint $A_i x_i \le y_i^\star$, problem (9) may be feasible or not. Figure 1 (left) shows an example with $z_i \in \mathbb{Z} \times \mathbb{R}$, in which (9) is feasible.

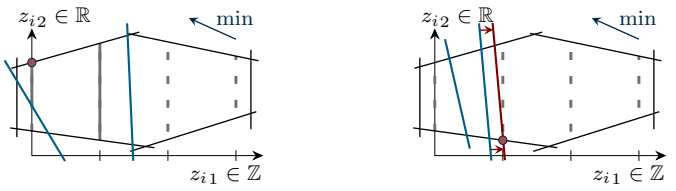

Fig. 1. Black lines represent P_i while the blue lines are the two components of $A_i z_i \le y_i^\star$. The resulting feasible set is denoted with solid vertical lines. Left: the problem is feasible and admits an optimal solution (red dot), which is mixed integer. Right: the problem is infeasible and the constraint $A_i z_i \le y_i^\star$ is enlarged just enough to include the closest mixed-integer vector (red dot).

In view of Proposition II.1, at least $N - S$ portions $z_i^\star$ of optimal solution are already mixed-integer and thus optimal for the corresponding local MILPs (9). Recall that $N \gg S$, thus *the majority* of subproblems (9) are feasible, while the total number of (possibly) infeasible instances is *at most* S. If (9) is infeasible for some agent i, we let the agent find a mixed-integer vector with the *minimal* violation of the allocation constraint as depicted in Figure 1 (right).

The procedure just outlined for the asymptotic case is entirely encoded in problem (8). To see this, let us now show how to operatively solve (8) with $y_i^\star$ in place of $y_i^{T_f}$. First, the needed violation of the allocation constraint is determined by computing ρ_i^∞ as the optimal cost of

$$\begin{aligned} \min_{\rho_i, x_i} \ & \rho_i \\ \text{subj. to } & A_i x_i \le y_i^\star + \rho_i \mathbb{1} \\ & x_i \in X_i^{\text{MILP}}, \ \ \rho_i \ge 0. \end{aligned} \tag{10}$$

Then, the value of ρ_i is fixed to ρ_i^∞ and problem (10) is re-optimized with its cost function replaced by $c_i^\top x_i$ to compute x_i^∞. If problem (9) is feasible, then $\rho_i^\infty = 0$ and the procedure is equivalent to solving (9). Instead, if problem (9) is infeasible, a violation $\rho_i^\infty \mathbb{1}$ of the allocation constraint is permitted.

Due to the violations, the aggregate mixed-integer solution $(x_1^\infty, \dots, x_N^\infty)$ may exceed the *restricted* total resource $b - \sigma$. Indeed, although it may not hold $\sum_{i=1}^N A_i x_i^\infty \le b - \sigma$, in the next subsection we show how to design the restriction σ to ensure that our original goal $\sum_{i=1}^N A_i x_i^\infty \le b$ is achieved.

Now, let us discuss how this approach can be adapted to cope with the finite number of iterations. The local allocation y_i^t can be thought of as $y_i^t = y_i^\star + \Delta_i^t$, where $\{\Delta_i^t\}_{t \ge 0} \to 0$ as t goes to infinity. By looking at problem (10), it is natural to expect that $\rho_i^t \le \rho_i^\infty + \Delta_i^t$. Thus, we let *all the agents* perform step (8) (implemented as in the asymptotic case). By employing an additional (small) restriction, we can guarantee that – after a sufficiently large time – the total violation is embedded into the restriction. A detailed analysis of this approach is given in Section IV-B.

D. Design of the Coupling Constraint Restriction

As already discussed, the purpose of the restriction σ is to compensate for possible violations of problem (8). Intuitively, we wish to make σ as small as possible for two reasons: the larger is σ, then (i) the more likely is (2) to be infeasible, (ii) the higher is the cost of the optimal solution $(z_1^\star, \dots, z_N^\star)$ of (2), which in turn deteriorates the cost of $(x_1^\infty, \dots, x_N^\infty)$.

We now propose a method to find a small a-priori restriction to guarantee feasibility of the computed solution. As before, we focus on the asymptotic case, while the extension to finite time is given in the following sections. Intuitively, the restriction must take into account the worst-case violation due to the mismatch between $(x_1^\infty, \dots, x_N^\infty)$ and $(z_1^\star, \dots, z_N^\star)$. Such worst case occurs when all the (at most S) agents for which $z_i^\star \notin X_i^{\text{MILP}}$ have infeasible instances of (9), leading to a positive violation $\rho_i^\infty > 0$. Thus, we define the a-priori restriction $\sigma^\infty \in \mathbb{R}^S$ as

$$\sigma^\infty = S \cdot \max_{i \in \{1,\dots,N\}} \sigma_i^{\text{LOC}}, \tag{11}$$

where $\sigma_i^{\text{LOC}} \in \mathbb{R}^S$ is the worst-case violation of agent i and $\max$ is intended component wise (we stick to this convention from now on).

Let us now quantify σ_i^{LOC}. Since $\text{conv}(X_i^{\text{MILP}})$ is bounded, it is possible to find a lower-bound vector, which we denote by $L_i \in \mathbb{R}^S$, for any *admissible* local allocation y_i,

$$L_i \triangleq \min_{x_i \in \text{conv}(X_i^{\text{MILP}})} A_i x_i = \min_{x_i \in X_i^{\text{MILP}}} A_i x_i, \tag{12}$$

By construction, it holds $L_i \le A_i z_i^\star \le y_i^\star$. Recall that each agent computes the needed violation through problem (10). Then, the worst-case violation that may occur at steady-state is $\rho_i^{\text{MAX}} \mathbb{1}$, where we define ρ_i^{MAX} as

$$\rho_i^{\text{MAX}} \triangleq \min_{x_i \in X_i^{\text{MILP}}} \ \max_{s \in \{1,\dots,S\}} [A_i x_i - L_i]_s, \tag{13}$$

where the notation $[\cdot]_s$ denotes the s-th component of a vector. Note that the optimization in (13) allows each agent i to find the "first" feasible vector, i.e., with minimal resource usage.

In order to reduce possible conservativeness of the violation, which can occur when $\rho_i^{\text{MAX}} > \max_{x_i \in X_i^{\text{MILP}}} [A_i x_i - L_i]_s$ for some component s of the coupling constraint, the computation of σ_i^{LOC} can be replaced by the saturated version

$$\sigma_i^{\text{LOC}} = \min \Big\{ \rho_i^{\text{MAX}} \mathbb{1}, \max_{x_i \in X_i^{\text{MILP}}} (A_i x_i - L_i) \Big\}.$$

In numerical computations, we have found that usually $\rho_i^{\text{MAX}} \ll \max_{x_i \in X_i^{\text{MILP}}} [A_i x_i - L_i]_s$, leading to $\sigma_i^{\text{LOC}} = \rho_i^{\text{MAX}} \mathbb{1}$.

We point out that the computation of σ^∞ must be performed in the initialization phase, which can be also carried out in a fully distributed way by using a max-consensus algorithm. In Figure 2, we illustrate an example of the restriction.

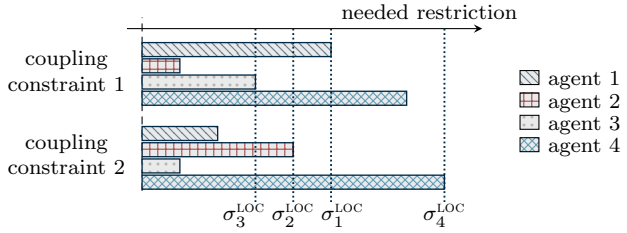

Fig. 2. Graphical representation of the restriction for $N = 4$ agents and $S = 2$ coupling constraints. Each bar represents the quantity $[A_i x_i - L_i]_s$ at an optimal solution of (13), which is the violation associated to the first feasible vector with minimal resource usage. The dotted lines represent ρ_i^{MAX} for all i, which here coincide with σ_i^{LOC}.

Remark III.3. *In [23], an alternative approach based on dual decomposition is explored to compute a feasible solution for MILP* (1)*. The restriction proposed in [23] is*

$$\sigma^{\text{DD}} = S \cdot \max_{i \in \{1,\dots,N\}} \max_{x_i \in X_i^{\text{MILP}}} (A_i x_i - L_i), \tag{14}$$

The term $\max_{x_i \in X_i^{\text{MILP}}} (A_i x_i - L_i)$ *may be overly conservative and in our approach it is replaced with* σ_i^{LOC}*, which can be thought of as the resource utilization of a feasible vector* closest *to* L_i*. Independently of the problem at hand, it holds* $\sigma^\infty \leq \sigma^{\text{DD}}$*.* □

IV. Analysis of Distributed Algorithm

In this section we provide both asymptotic and finite-time analyses of Algorithm 1 under the following assumption.

Assumption IV.1. *For a given* $\sigma \geq 0$*, the optimal solution of problem* (2) *is unique.* □

This assumption ensures that the optimal solution of (2) is a vertex (hence Proposition II.1 applies). It can be guaranteed by simply adding a small, random perturbation to the cost vectors c_i. Notice that Assumption IV.1 is also needed in dual decomposition approaches such as [23]–[25]. Notably, dual decomposition approaches require also uniqueness of the dual optimal solution of problem (2), while our approach is less restrictive since this is not necessary.

A. Asymptotic analysis

First, we proceed under the assumption that $\sigma = \sigma^\infty$ as in (11) and that the algorithm is executed until convergence to an optimal allocation $(y_1^\star, \dots, y_N^\star)$, i.e., an optimal solution of problem (3). Indeed, steps (6)–(7) implement the distributed algorithm in Section III-A for the solution of problem (2), so that Proposition III.2 (ii) applies.

The next theorem shows feasibility of the computed mixed-integer solution for the target MILP (1).

Theorem IV.2 (Feasibility)**.** *Let* $\sigma = \sigma^\infty$ *as in* (11)*, and let problem* (2) *be feasible and satisfy Assumption IV.1. Let* $(y_1^\star, \dots, y_N^\star)$ *be an optimal solution of problem* (3)*. Then, the vector* $(x_1^\infty, \dots, x_N^\infty)$*, with each* x_i^∞ *optimal solution of* (8) *with* $y_i^t = y_i^\star$*, is feasible for MILP* (1)*, i.e.,* $x_i^\infty \in X_i^{\text{MILP}}$ *for all* $i \in \{1, \dots, N\}$ *and* $\sum_{i=1}^N A_i x_i^\infty \leq b$*.* □

The proof of Theorem IV.2 is given in appendix.

Remark IV.3. *The proof of Theorem IV.2 reveals that the same result can be obtained by using an allocation* $(y_1, \dots, y_N)$ *associated to* any *vertex of the feasible set of problem* (2) *(rather than an optimal allocation* $(y_1^\star, \dots, y_N^\star)$*). Proposition II.1 can still be applied and the proof remains unchanged.* □

Theorem IV.2 guarantees that the computed solution is feasible for the target MILP (1), but, in general, there is a certain degree of suboptimality. In the following, we provide suboptimality bounds under Slater's constraint qualification.

Assumption IV.4. *For a given* $\sigma > 0$*, there exists a vector* $(\widehat{z}_1, \dots, \widehat{z}_N)$*, with* $\widehat{z}_i \in \text{conv}(X_i^{\text{MILP}})$ *for all* i*, such that*

$$\zeta \triangleq \min_{s \in \{1,\dots,S\}} \Big[b - \sigma - \sum_{i=1}^N A_i \widehat{z}_i \Big]_s > 0. \tag{15}$$

The cost of $(\widehat{z}_1, \dots, \widehat{z}_N)$ *is denoted by* $J^{\text{SL}} = \sum_{i=1}^N c_i^\top \widehat{z}_i$*.* □

The following result establishes an a-priori suboptimality bound on the mixed-integer solution with $\sigma = \sigma^\infty$ as in (11). Due to space constraints, the proofs of Theorem IV.5 and Corollary IV.6 are omitted. The reader is referred to [23] for similar results.

Theorem IV.5 (A-Priori Suboptimality Bound)**.** *Consider the same assumptions and quantities of Theorem IV.2 and let also Assumption IV.4 hold. Then,* $(x_1^\infty, \dots, x_N^\infty)$ *satisfies the suboptimality bound* $\sum_{i=1}^N c_i^\top x_i^\infty - J^{\text{MILP}} \leq B$*, where* J^{MILP} *is the optimal cost of* (1) *and* B *is defined as*

$$B \triangleq \left(S + \frac{N \|\sigma^\infty\|_\infty}{\zeta} \right) \max_{i \in \{1,\dots,N\}} \gamma_i,$$

with ζ *defined in* (15)*, and* $\gamma_i \triangleq \max_{x_i \in X_i^{\text{MILP}}} c_i^\top x_i - \min_{x_i \in X_i^{\text{MILP}}} c_i^\top x_i$*.* □

Note that, although the bound provided by Theorem IV.5 is formally analogous to [23, Theorem 3.3], there is an implicit difference due to the restriction values (cf. Remark III.3). In particular, our bound is tighter since σ^∞ is less than or equal to the restriction proposed by [23].

A tighter bound can be derived by using the steady-state solution of the algorithm and computing also the primal solution of (6) for all i. For this reason, we call this bound a posteriori, since it depends on the solution computed by Algorithm 1.

Corollary IV.6 (A-Posteriori Suboptimality Bound)**.** *Consider the same assumptions and quantities of Theorem IV.5. Then,* $\sum_{i=1}^N c_i^\top x_i^\infty - J^{\text{MILP}} \leq B'$*, where* B' *is defined as*

$$B' \triangleq \sum_{i \in I_{\mathbb{R}}} (c_i^\top x_i^\infty - c_i^\top z_i^\star) + \frac{\|\sigma^\infty\|_\infty}{\zeta} (J^{\text{SL}} - \sum_{i=1}^N c_i^\top z_i^\star), \tag{16}$$

with ζ and J^{SL} defined in Assumption IV.4 and $I_{\mathbb{R}}$ containing the indices of agents such that $z_i^\star \notin X_i^{\text{MILP}}$ ($|I_{\mathbb{R}}| \leq S$). □

B. Finite-time Analysis

In this section, we provide a finite-time analysis of the distributed algorithm. All the proofs of this subsection are given in appendix. To this end, we assume that the restriction is equal to an enlarged version of the asymptotic restriction in (11), i.e.,

$$\sigma^{\text{FT}} = \sigma^{\infty} + \delta \mathbb{1}, \tag{17}$$

for an arbitrary $\delta > 0$. We assume problem (2) is feasible with this new restriction. We provide two results that extend the results of Section IV-A to a finite-time setting.

At a high level, finite-time feasibility hinges upon the fact that the allocation sequence $\{y_1^t, \dots, y_N^t\}_{t\geq 0}$ approaches an optimal allocation. Eventually, the additional restriction δ can embed the distance of the current allocation estimate to optimality. The next theorem formalizes this result.

Theorem IV.7 (Finite-time feasibility)**.** *Let $\sigma = \sigma^{\text{FT}}$ as in* (17)*, for some $\delta > 0$, and let problem* (2) *be feasible and satisfy Assumption IV.1. Consider the mixed-integer sequence $\{x_1^t, \dots, x_N^t\}_{t\geq 0}$ generated by Algorithm 1 under Assumption III.1, with $\sum_{i=1}^N y_i^0 = b - \sigma^{\text{FT}}$. There exists a sufficiently large time $T_\delta > 0$ such that the vector $(x_1^t, \dots, x_N^t)$ is a feasible solution for problem* (1)*, i.e., $x_i^t \in X_i^{\text{MILP}}$ for all $i \in \{1, \dots, N\}$ and $\sum_{i=1}^N A_i x_i^t \leq b$, for all $t \geq T_\delta$.* □

In principle, the smaller is δ, the longer it takes for the mixed-integer vector $(x_1^t, \dots, x_N^t)$ to satisfy the coupling constraint. As a function of δ, there is a trade-off between the number of iterations to guarantee solution feasibility and how strict is the assumption that problem (2) is feasible.

Next, we provide a suboptimality bound that can be evaluated when the algorithm is halted.

Theorem IV.8 (Finite-time suboptimality bound)**.** *Consider the same assumptions and quantities of Theorem IV.7 and let also Assumption IV.4 hold. Moreover, let $\epsilon_i > 0$ for $i \in \{1, \dots, N\}$. Then, there exists a time $T_\epsilon > 0$ such that the vector $(x_1^t, \dots, x_N^t)$ satisfies the suboptimality bound $\sum_{i=1}^N c_i^\top x_i^t - J^{\text{MILP}} \leq B^t$ for all $t \geq T_\epsilon$, with B^t being*

$$B^t \triangleq \sum_{i=1}^N (c_i^\top x_i^t - J_i^{\text{LP},t}) + \sum_{i=1}^N \epsilon_i \|\mu_i^t\|_1 + \Gamma \|\sigma^{\text{FT}}\|_\infty, \tag{18}$$

where J^{MILP} is the optimal cost of (1)*, $J_i^{\text{LP},t}$ and μ_i^t are the optimal cost and a Lagrange multiplier of* (6) *at time t, $\Gamma = \frac{N}{\zeta} \sum_{i=1}^N \Big(\max_{x_i \in X_i^{\text{MILP}}} c_i^\top x_i - \min_{x_i \in X_i^{\text{MILP}}} c_i^\top x_i \Big)$, and ζ associated to any Slater point (cf. Assumption IV.4).* □

We point out that, differently from the asymptotic case, the bound (18) does not depend on the solution of (6), but only on its optimal cost. Moreover, notice that the bound (18) is a posteriori, while if an a-priori bound with restriction σ^{FT} is desired, it still has the form of Theorem IV.5 (since it does not depend on the algorithmic evolution).

V. Monte Carlo Numerical Computations

In this section, we provide a computational study on randomly generated MILPs to compare Algorithm 1 with [23]. The distributed algorithms are emulated using a single machine with the MATLAB software and the local MILPs are solved using the integrated numerical solver.

We consider large-scale problems with a total of 4500 optimization variables (3000 are integer and 1500 are continuous). There are $N = 300$ agents and $S = 5$ coupling constraints. The local constraints X_i^{MILP} are subsets of $\mathbb{Z}^{10} \times \mathbb{R}^5$ satisfying $D_i x_i \leq d_i$, where D_i and $d_i \in \mathbb{R}^{20}$ have random entries in $[0, 1]$ and $[20, 40]$ respectively. Box constraints $-60\,\mathbb{1} \leq x_i \leq 60\,\mathbb{1}$ are added to ensure compactness. The cost vector is $c_i = D_i^\top \hat{c}_i$, where $\hat{c}_i$ has random entries in $[0, 5]$. As for the coupling, the matrices A_i are random with entries in $[0, 1]$ and the resource vector $b \in \mathbb{R}^5$ is random with values in two different intervals. Specifically, we first pick values in $[-20N, -15N]$, which results in a "loose" b, then we pick values in $[-180N, -175N]$, which results in a "tight" b.

A total of 100 MILPs with loose b and 100 MILPs with tight b are generated. For each problem, we check feasibility of problem (2) for both our method and the method in [23]. Then, both algorithms are executed up to asymptotic convergence to evaluate the mixed-integer solution suboptimality. The results are summarized in Figures 3 and 4, where the restriction size is computed as $\|\sigma\|/\|b\|$ and the suboptimality is computed as $\big(\sum_{i=1}^N c_i^\top x_i^\infty - q^\star\big)/q^\star$, with $q^\star$ being the optimal cost of (2). The number of solvable instances is the number of problems for which problem (2) is feasible. For loose b, both methods are always applicable. However, our approach provides better solution performance than [23]. For tight resource vectors, our method is still applicable in the 70% of the cases, and provides an average suboptimality of 6.91%, while the approach in [23] cannot be applied due to infeasibility of the approximate problem (2) (caused by the too large needed restrictions). It is worth noting that the generation interval $[-180N, -175N]$ cannot be further tightened. Indeed, for smaller values of b, the target MILP (1) becomes infeasible.

Finally, we show the evolution of Algorithm 1 on a single instance over an Erdős-Rényi graph with edge probability 0.2. Figure 5 shows the value of the coupling constraints along the algorithmic evolution, with $\delta = 0.5$ (cf. (17)). Note that feasibility is achieved in finite time (within 400 iterations), confirming Theorem IV.7. In order to detect feasibility, agents can run a consensus-based scheme from time to time to check whether the current solution satisfies the coupling constraints.

	Algorithm 1		[23]	
	b loose	b tight	b loose	b tight
# solvable problems	100%	70%	100%	0%
size of restriction	7.4%	0.72%	66.9%	6.63%
suboptimality of solution	0.06%	6.91%	0.7%	N.A.

Fig. 3. Montecarlo simulations on random MILPs: performance comparison of Algorithm 1 and of the method in [23]. The second and the third row are averaged over the Monte Carlo runs. See the text for further details.

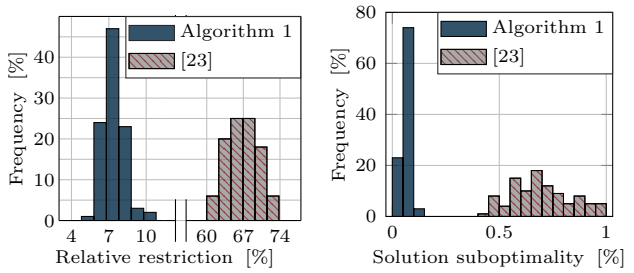

Fig. 4. Montecarlo simulations on random MILPs with loose b: comparison histograms of Algorithm 1 and of the method in [23]. See the text for details.

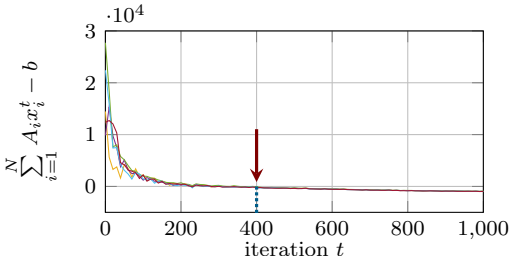

Fig. 5. Components of the coupling constraint associated to $(x_1^t, \dots, x_N^t)$ generated by Algorithm 1 on a random MILP. The agents are assumed to compute the mixed-integer solution according to (8) at each iteration.

VI. Conclusions

In this paper we proposed a distributed algorithm to compute a feasible solution to large-scale MILPs with high optimality degree. The method is based on a primal decomposition approach and a suitable restriction of the coupling constraint and guarantees feasibility of the computed mixed-integer vectors in finite time. Asymptotic and finite-time results for feasibility and suboptimality bounds are proved. Numerical simulations highlight the efficacy of the proposed methodology.

References

[1] A. Camisa, I. Notarnicola, and G. Notarstefano, "A primal decomposition method with suboptimality bounds for distributed mixed-integer linear programming," in *IEEE Conference on Decision and Control*, 2018, pp. 3391–3396.

[2] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[3] Y. Kuwata and J. P. How, "Cooperative distributed robust trajectory optimization using receding horizon MILP," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 2, pp. 423–431, 2010.

[4] M. A. Müller, M. Reble, and F. Allgöwer, "A general distributed MPC framework for cooperative control," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 7987–7992, 2011.

[5] Q. Tran-Dinh, I. Necoara, and M. Diehl, "A dual decomposition algorithm for separable nonconvex optimization using the penalty function framework," in *IEEE Conference on Decision and Control (CDC)*, 2013, pp. 2372–2377.

[6] P. Giselsson, M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer, "Accelerated gradient methods and dual decomposition in distributed Model Predictive Control," *Automatica*, vol. 49, no. 3, pp. 829–833, 2013.

[7] Z. Wang and C.-J. Ong, "Accelerated distributed MPC of linear discrete-time systems with coupled constraints," *IEEE Transactions on Automatic Control*, vol. 63, no. 11, pp. 3838–3849, 2018.

[8] I. Necoara, V. Nedelcu, and I. Dumitrache, "Parallel and distributed optimization methods for estimation and control in networks," *Journal of Process Control*, vol. 21, no. 5, pp. 756–766, 2011.

[9] H. Lakshmanan and D. P. De Farias, "Decentralized resource allocation in dynamic networks of agents," *SIAM Journal on Optimization*, vol. 19, no. 2, pp. 911–940, 2008.

[10] A. Simonetto, T. Keviczky, and M. Johansson, "A regularized saddle-point algorithm for networked optimization with resource allocation constraints," in *IEEE Conf. on Decision and Control (CDC)*, 2012, pp. 7476–7481.

[11] A. Cherukuri and J. Cortés, "Distributed generator coordination for initialization and anytime optimization in economic dispatch," *IEEE Trans. on Control of Network Systems*, vol. 2, no. 3, pp. 226–237, 2015.

[12] ——, "Initialization-free distributed coordination for economic dispatch under varying loads and generator commitment," *Automatica*, vol. 74, pp. 183–193, 2016.

[13] A. Simonetto and H. Jamali-Rad, "Primal recovery from consensus-based dual decomposition for distributed convex optimization," *Journal of Optim. Theory and Applications*, vol. 168, no. 1, pp. 172–197, 2016.

[14] A. Falsone, K. Margellos, S. Garatti, and M. Prandini, "Dual decomposition for multi-agent distributed optimization with coupling constraints," *Automatica*, vol. 84, pp. 149–158, 2017.

[15] I. Notarnicola and G. Notarstefano, "Constraint-coupled distributed optimization: A relaxation and duality approach," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 1, pp. 483–492, 2020.

[16] Y. Zhang and M. M. Zavlanos, "A consensus-based distributed augmented Lagrangian method," in *IEEE Conference on Decision and Control (CDC)*, 2018, pp. 1763–1768.

[17] A. Falsone, I. Notarnicola, G. Notarstefano, and M. Prandini, "Tracking-ADMM for distributed constraint-coupled optimization," *Automatica*, vol. 117, p. 108962, 2020.

[18] G. Notarstefano, I. Notarnicola, and A. Camisa, "Distributed optimization for smart cyber-physical networks," *Foundations and Trends® in Systems and Control*, vol. 7, no. 3, pp. 253–383, 2019.

[19] S.-J. Kim and G. B. Giannakis, "Scalable and robust demand response with mixed-integer constraints," *IEEE Transactions on Smart Grid*, vol. 4, no. 4, pp. 2089–2099, 2013.

[20] R. Takapoui, N. Moehle, S. Boyd, and A. Bemporad, "A simple effective heuristic for embedded mixed-integer quadratic programming," *International Journal of Control*, pp. 1–11, 2017.

[21] Y. Kuwata and J. P. How, "Cooperative distributed robust trajectory optimization using receding horizon MILP," *IEEE Transctions on Control Systems Technology*, vol. 19, no. 2, pp. 423–431, 2011.

[22] A. Testa, A. Rucco, and G. Notarstefano, "Distributed mixed-integer linear programming via cut generation and constraint exchange," *IEEE Transactions on Automatic Control*, vol. 65, no. 4, pp. 1456–1467, 2019.

[23] R. Vujanic, P. M. Esfahani, P. J. Goulart, S. Mariéthoz, and M. Morari, "A decomposition method for large scale MILPs, with performance guarantees and a power system application," *Automatica*, vol. 67, no. 5, pp. 144–156, 2016.

[24] A. Falsone, K. Margellos, and M. Prandini, "A decentralized approach to multi-agent MILPs: finite-time feasibility and performance guarantees," *Automatica*, vol. 103, pp. 141–150, 2019.

[25] ——, "A distributed iterative algorithm for multi-agent MILPs: finite-time feasibility and performance characterization," *IEEE Control Systems Letters*, vol. 2, no. 4, pp. 563–568, 2018.

[26] D. P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*. Academic Press, 1982.

[27] G. J. Silverman, "Primal decomposition of mathematical programs by resource allocation: I–basic theory and a direction-finding procedure," *Operations Research*, vol. 20, no. 1, pp. 58–74, 1972.

[28] D. P. Bertsekas, *Nonlinear programming*. Athena scientific, 1999.

[29] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

Appendix

A. Proof of Theorem IV.2

For the sake of analysis, let us denote by $(z_1^\star, \dots, z_N^\star)$ the optimal solution of the restricted LP (2). By Assumption IV.1, $(z_1^\star, \dots, z_N^\star)$ is a vertex, so that by Proposition II.1 there exists $I_\mathbb{Z} \subseteq \{1, \dots, N\}$, with $|I_\mathbb{Z}| \geq N - S$, such that $z_i^\star \in X_i^{\text{MILP}}$ for all $i \in I_\mathbb{Z}$. By Lemma II.2, $z_i^\star$ is an optimal solution of problem (4), with $y_i = y_i^\star$, for all $i \in \{1, \dots, N\}$. Thus, for all $i \in I_\mathbb{Z}$, $z_i^\star \in X_i^{\text{MILP}}$ is the optimal solution of (8) with $y_i^t = y_i^\star$. Then, it holds $A_i x_i^\infty \leq y_i^\star$ for all $i \in I_\mathbb{Z}$.

Let us focus on the set $I_{\mathbb{R}} = \{1,\dots,N\}\setminus I_{\mathbb{Z}}$, which contains indices such that $z_i^\star \notin X_i^{\text{MILP}}$. Let us further partition $I_{\mathbb{R}} = I_{\text{FEAS}} \cup I_{\text{INFEAS}}$, where the indices collected in I_{FEAS} correspond to feasible subproblems (4), from which it follows that $A_i x_i^\infty \leq y_i^\star$ for all $i \in I_{\text{FEAS}}$, while the remaining index set I_{INFEAS} corresponds to infeasible subproblems (4). We have

$$A_i x_i^\infty \overset{(a)}{\leq} y_i^\star + \rho_i^\infty \mathbb{1} \overset{(b)}{\leq} y_i^\star + \rho_i^{\text{MAX}} \mathbb{1}, \qquad \forall\, i \in I_{\text{INFEAS}},$$

where *(a)* follows by construction of x_i^∞ and *(b)* follows since any optimal solution of problem (13), say x_i^{L}, is feasible for problem (10) (since $A_i x_i^{\text{L}} \leq L_i + \rho_i^{\text{MAX}} \leq y_i^\star + \rho_i^{\text{MAX}}$), from which it follows that $\rho_i^\infty \leq \rho_i^{\text{MAX}}$ (by optimality). Also, notice that, since $x_i^\infty \in X_i^{\text{MILP}}$, then $[A_i x_i^\infty]_s \leq \max_{x_i \in X_i^{\text{MILP}}} [A_i x_i]_s$ for all s and it holds $A_i x_i^\infty - y_i^\star \leq \max_{x_i \in X_i^{\text{MILP}}} A_i x_i - y_i^\star \leq \max_{x_i \in X_i^{\text{MILP}}} A_i x_i - L_i$, where $\max$ is intended component wise. It follows that $A_i x_i^\infty \leq y_i^\star + \sigma_i^{\text{LOC}}$ for all $i \in I_{\text{INFEAS}}$. By summing over $i \in I_{\text{INFEAS}}$ the term σ_i^{LOC}, we obtain

$$\begin{aligned}\sum_{i\in I_{\text{INFEAS}}} \sigma_i^{\text{LOC}} &\leq |I_{\text{INFEAS}}| \max_{i\in I_{\text{INFEAS}}} \sigma_i^{\text{LOC}} \\ &\leq S \max_{i\in\{1,\dots,N\}} \sigma_i^{\text{LOC}} = \sigma^\infty, \end{aligned} \tag{19}$$

where $\max$ is intended component wise. Collecting the previous conditions leads to

$$\begin{aligned}\sum_{i=1}^N A_i x_i^\infty &= \sum_{i\in I_{\mathbb{Z}}} A_i x_i^\infty + \sum_{i\in I_{\text{FEAS}}} A_i x_i^\infty + \sum_{i\in I_{\text{INFEAS}}} A_i x_i^\infty \\ &\leq \sum_{i=1}^N y_i^\star + \sum_{i\in I_{\text{INFEAS}}} \sigma_i^{\text{LOC}} \leq b - \sigma^\infty + \sigma^\infty = b,\end{aligned}$$

where we used $\sum_{i=1}^N y_i^\star = b - \sigma^\infty$. The proof follows. □

B. Proof of Theorem IV.7

Let $\{y_1^t,\dots,y_N^t\}_{t\geq 0}$ denote the allocation vector sequence generated by Algorithm 1. By Proposition III.2, the sequence $\{y_1^t,\dots,y_N^t\}_{t\geq 0}$ converges to an optimal solution $(y_1^\star,\dots,y_N^\star)$ of problem (3) with $\sigma = \sigma^\infty + \delta\mathbb{1}$. Thus, for all $i \in \{1,\dots,N\}$ and $\epsilon_i > 0$, there exists $T_{\epsilon_i} > 0$ such that $t \geq T_{\epsilon_i} \Rightarrow \|y_i^t - y_i^\star\|_\infty \leq \epsilon_i$. If we let $T = \max_{i\in\{1,\dots,N\}} T_{\epsilon_i}$, then $y_i^t \leq y_i^\star + \epsilon_i \mathbb{1}$ for all $t \geq T$ and $i \in \{1,\dots,N\}$.

To prove the statement, we compare the state of the algorithm at an iteration $t \geq T_\delta$ and the quantities that would be computed at infinity, for all $i \in \{1,\dots,N\}$. To this end, let us denote by $(\rho_i^\infty, x_i^\infty)$ the optimal solution of problem (8) with $y_i^t = y_i^\star$ (we discard the ξ_i part of the solution). As shown in Section III-C, in order to compute ρ_i^t, agents can equivalently solve

$$\begin{aligned}\min_{\rho_i, x_i}\ & \rho_i \\ \text{subj. to}\ & A_i x_i \leq y_i^t + \rho_i \mathbb{1} \\ & x_i \in X_i^{\text{MILP}},\ \ \rho_i \geq 0.\end{aligned} \tag{20}$$

The pair $(\epsilon_i + \rho_i^\infty, x_i^\infty)$ is feasible for problem (20) for all $t \geq T$. Indeed, it holds $x_i^\infty \in X_i$, $\epsilon_i + \rho_i^\infty \geq 0$, and moreover $A_i x_i^\infty \leq y_i^\star + \rho_i^\infty \mathbb{1} \leq y_i^t + \epsilon_i \mathbb{1} + \rho_i^\infty \mathbb{1}$ for all $t \geq T$. Being ρ_i^t the optimal cost of (20), we have $\rho_i^t \leq \epsilon_i + \rho_i^\infty$ for all $t \geq T$.

We now follow arguments similar to the proof of Theorem IV.2. For all $i \in \{1,\dots,N\}$, let $z_i^\star$ denote the optimal solution of problem (2). For the sake of analysis, let us split the agent set $\{1,\dots,N\}$ as $I_{\mathbb{Z}} \cup I_{\text{FEAS}} \cup I_{\text{INFEAS}}$, where $I_{\mathbb{Z}}$ contains agents for which $z_i^\star \in X_i^{\text{MILP}}$, I_{FEAS} contains agents for which $z_i^\star \notin X_i^{\text{MILP}}$ and $\rho_i^\infty = 0$, and I_{INFEAS} contains agents for which $z_i^\star \notin X_i^{\text{MILP}}$ and $\rho_i^\infty > 0$. Using the same arguments of Theorem IV.2, for all $i \in I_{\mathbb{Z}}$ it holds $\rho_i^\infty = 0$.

Now, let us consider the agents $i \in I_{\mathbb{Z}} \cup I_{\text{FEAS}}$. By construction, it holds

$$A_i x_i^t \leq y_i^t + \rho_i^t \leq y_i^t + \epsilon_i,\ \forall\, i \in I_{\mathbb{Z}} \cup I_{\text{FEAS}} \text{ and } t \geq T_\delta, \tag{21}$$

where we also used $\rho_i^t \leq \epsilon_i + \rho_i^\infty$. As for the agents $i \in I_{\text{INFEAS}}$, again by $\rho_i^t \leq \epsilon_i + \rho_i^\infty$, it holds $A_i x_i^t - y_i^t \leq \rho_i^t \mathbb{1} \leq \rho_i^\infty \mathbb{1} + \epsilon_i \mathbb{1}$ for all $t \geq T_\delta$, or equivalently $A_i x_i^t - y_i^t - \epsilon_i \mathbb{1} \leq \rho_i^\infty \mathbb{1}$, for all $t \geq T_\delta$. Moreover, note that, for $t \geq T_\delta$, it holds $A_i x_i^t - y_i^t - \epsilon_i \mathbb{1} \leq A_i x_i^t - y_i^t \leq \max_{x_i \in X_i^{\text{MILP}}} A_i x_i - L_i$, where $\max$ is intended component wise. Using the definition of σ_i^{LOC} in Section III-D and rearranging the terms, we obtain

$$A_i x_i^t \leq y_i^t + \sigma_i^{\text{LOC}} + \epsilon_i \mathbb{1}, \qquad \forall\, i \in I_{\text{INFEAS}} \text{ and } t \geq T_\delta. \tag{22}$$

Finally, by using (21) and (22), we can write

$$\begin{aligned}\sum_{i=1}^N A_i x_i^t &= \sum_{i=1}^N (y_i^t + \epsilon_i \mathbb{1}) + \sum_{i\in I_{\text{INFEAS}}} \sigma_i^{\text{LOC}} \\ &\leq b - \sigma^\infty - \delta\mathbb{1} + \sum_{i=1}^N \epsilon_i \mathbb{1} + \sigma^\infty, \qquad \forall t \geq T,\end{aligned}$$

which follows since $\sum_{i=1}^N y_i^t = \sum_{i=1}^N y_i^0 = b - \sigma^\infty - \delta\mathbb{1}$ (cf. also Proposition III.2 (i)) and by (19). Since ϵ_i are arbitrary, choosing $\epsilon_i = \delta/N$ for all i implies $\sum_{i=1}^N A_i x_i^t \leq b$ for all $t \geq T_\delta \triangleq T$, which concludes the proof. □

C. Proof of Theorem IV.8

Let $\{y_1^t,\dots,y_N^t\}_{t\geq 0}$ denote the allocation vector sequence generated by Algorithm 1. By following similar arguments as in the proof of Theorem IV.7, we conclude that, for fixed $\epsilon_i > 0$, there exists a sufficiently large $T > 0$ such that $\|y_i^\star - y_i^t\|_\infty \leq \epsilon_i$ for all $t \geq T$ and $i \in \{1,\dots,N\}$.

As done in [23, Theorem 3.3], let us split the suboptimality bound as $\sum_{i=1}^N c_i^\top x_i^t - J^{\text{MILP}} = \sum_{i=1}^N (c_i^\top x_i^t - J_i^{\text{LP},t}) + \big(\sum_{i=1}^N J_i^{\text{LP},t} - J^{\text{LP},\sigma^{\text{FT}}}\big) + \big(J^{\text{LP},\sigma^{\text{FT}}} - J^{\text{LP}}\big) + \big(J^{\text{LP}} - J^{\text{MILP}}\big)$, where $J^{\text{LP},\sigma^{\text{FT}}}$ denotes the optimal cost of problem (2) with $\sigma = \sigma^{\text{FT}}$. The first term $\sum_{i=1}^N (c_i^\top x_i^t - J_i^{\text{LP},t})$ can be explicitly computed. As for the last two terms, by following similar arguments as in [23], we conclude that $\big(J^{\text{LP},\sigma^{\text{FT}}} - J^{\text{LP}}\big) + \big(J^{\text{LP}} - J^{\text{MILP}}\big) \leq \Gamma \|\sigma^{\text{FT}}\|_\infty$. Let us analyze in detail the second term.

Notice that $J^{\text{LP},\sigma^{\text{FT}}}$ can be seen as the optimal cost of a perturbed version of the problem having optimal cost $\sum_{i=1}^N J_i^{\text{LP},t}$, namely the aggregate problem solved by the agents at iteration t, i.e.,

$$\begin{aligned}\min_{\substack{z_1,\dots,z_N,\\ v_1,\dots,v_N}}\ & \sum_{i=1}^N (c_i^\top z_i + M v_i) \\ \text{subj. to}\ & A_i z_i \leq y_i^t + v_i \mathbb{1}, && \forall\, i, \\ & z_i \in \text{conv}(X_i^{\text{MILP}}),\ \ v_i \geq 0, && \forall\, i,\end{aligned} \tag{23}$$

In particular, the constraints $A_i z_i \leq y_i^t + v_i \mathbb{1}$ are perturbed by $y_i^\star - y_i^t$ to obtain $A_i z_i \leq y_i^\star + v_i \mathbb{1}$. By applying perturbation theory [29], we have for all $t \geq T$

$$\sum_{i=1}^N J_i^{\text{LP},t} - J^{\text{LP},\sigma^{\text{FT}}} \leq \sum_{i=1}^N \|y_i^\star - y_i^t\|_\infty \|\mu_i^t\|_1 \leq \sum_{i=1}^N \epsilon_i \|\mu_i^t\|_1. \ \square$$